

Лабораторная работа №1, Часть 1: Подготовка рабочего окружения для MLOps

Цель работы: Освоить базовые принципы установки и настройки современного рабочего окружения для Data Science и MLOps, основанного на дистрибутиве Anaconda (conda) и технологии Docker. Получить практические навыки управления виртуальными окружениями Python, работы с менеджером пакетов conda и выполнения основных операций с Docker-контейнерами.

Стек технологий:

- **Операционная система:** Ubuntu 24.04 LTS
 - **Менеджер пакетов и окружений:** Conda (в составе дистрибутива Anaconda или Miniconda)
 - **Технология контейнеризации:** Docker
 - **Платформа для интерактивных вычислений:** JupyterLab
-

Теоретическая часть (краткое содержание)

1. Введение в Conda и виртуальные окружения Современная разработка на Python, особенно в области анализа данных и машинного обучения, требует управления множеством зависимостей (библиотек) и их версий. Виртуальное окружение — это изолированное пространство, позволяющее устанавливать конкретные версии пакетов, не влияя на глобальную установку Python или другие проекты. **Conda** — это кроссплатформенный менеджер пакетов и управления окружениями с открытым исходным кодом. В отличие от стандартного **venv**, conda может управлять не только Python-пакетами, но и бинарными зависимостями (например, библиотеками C/C++), что критически важно для работы таких пакетов, как NumPy, SciPy или TensorFlow. **Anaconda** — это дистрибутив Python и R, который включает в себя conda, множество предустановленных научных пакетов (data science stack) и графические утилиты. **Miniconda** — его минималистичная версия, содержащая только conda, Python и небольшой набор базовых пакетов.

2. Введение в Docker **Docker** — это платформа для разработки, доставки и запуска приложений в контейнерах. **Контейнер** — это стандартизированная единица программного обеспечения, которая инкапсулирует код и все его зависимости, обеспечивая быстрое и надежное выполнение приложения в любой среде (ноутбук, сервер, облако). Ключевые преимущества:

- **Изоляция:** Приложение в контейнере не зависит от окружения хостовой системы.
- **Переносимость:** Образ, собранный на одной машине, гарантированно запустится на другой.
- **Повторяемость:** Исключается проблема "но на моей машине это работало".
- **Масштабируемость:** Легко запустить несколько экземпляров приложения.

Базовые концепции:

- **Docker Image:** Шаблон только для чтения, используемый для создания контейнеров. Собирается из Dockerfile.
 - **Docker Container:** Запущенный экземпляр образа.
 - **Docker Hub:** Публичный реестр для хранения и обмена образами.
-

Задание на практическую реализацию

Этап 1: Установка и настройка Miniconda

1. Загрузка и установка Miniconda:

- Откройте терминал (**Ctrl+Alt+T**).
- Загрузите последнюю версию установщика Miniconda для Linux (x86_64) с официального сайта с помощью **wget**:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

- Запустите установщик (ответьте **yes** на запрос лицензии и запуска **conda init**):

```
bash Miniconda3-latest-Linux-x86_64.sh
```

- Закройте и снова откройте терминал, чтобы активировать изменения. В начале строки приглашения терминала должно появиться **(base)**, что указывает на активацию базового окружения conda.

2. Создание и активация виртуального окружения:

- Создайте новое виртуальное окружение с именем **mlops-lab** и Python версии 3.10:

```
conda create -n mlops-lab python=3.10
```

- Активируйте созданное окружение:

```
conda activate mlops-lab
```

- Убедитесь, что приглашение терминала изменилось с **(base)** на **(mlops-lab)**.

3. Установка пакетов в созданное окружение:

- Установите основные пакеты для Data Science и последующей работы с MLflow, используя менеджер пакетов **pip**:

```
pip install numpy pandas scikit-learn matplotlib jupyterlab
```

- Проверьте установку, например, попробовав импортировать **pandas**:

```
python -c "import pandas as pd; print(pd.__version__)"
```

Этап 2: Установка Docker и базовые операции

1. Установка Docker Engine:

- Обновите индексы пакетов и установите необходимые зависимости:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
```

- Добавьте официальный GPG-ключ Docker:

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

- Настройте репозиторий:

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
  by=/etc/apt/keyrings/docker.asc]
  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Установите Docker Engine:

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

- Проверьте установку, запустив тестовый образ:

```
sudo docker run hello-world
```

Эта команда загрузит тестовый образ и запустит его в контейнере, который выведет сообщение об успешной установке.

2. Настройка прав для не-root пользователя (ОБЯЗАТЕЛЬНО):

- По умолчанию для управления Docker требуется права `sudo`. Чтобы запускать команды `docker` без `sudo`, добавьте своего пользователя в группу `docker`:

```
sudo usermod -aG docker $USER
```

- Закройте терминал и войдите в систему заново**, чтобы изменения вступили в силу.

3. Базовые команды Docker:

- Запуск контейнера:** Запустите контейнер из образа `ubuntu:24.04` в интерактивном режиме (`-it`):

```
docker run -it ubuntu:24.04 /bin/bash
```

Вы окажетесь внутри контейнера. Попробуйте выполнить команды `ls /`, `cat /etc/os-release`. Выйдите из контейнера командой `exit`.

- Просмотр запущенных контейнеров:** Команда `docker ps` покажет работающие контейнеры. Чтобы увидеть все контейнеры (включая остановленные), используйте `docker ps -a`.
- Просмотр образов:** Команда `docker images` отобразит список образов, скачанных на ваш компьютер.
- Удаление контейнера:** Найдите ID остановленного контейнера (`docker ps -a`) и удалите его:

```
docker rm <container_id>
```

4. Запуск JupyterLab в Docker-контейнере:

- Запустите предсобранный образ JupyterLab, предоставляемый проектом Jupyter Docker Stacks. Эта команда делает порт 8888 контейнера доступным на локальной машине и монтирует текущую рабочую директорию (`$PWD`) в директорию `/home/jovyan/work` внутри контейнера:

```
docker run -p 8888:8888 -v $PWD:/home/jovyan/work jupyter/datascience-notebook:latest
```

- В логах запуска найдите URL-адрес вида `http://127.0.0.1:8888/lab?token=....`. Скопируйте его и откройте в браузере.
- Вы должны увидеть интерфейс JupyterLab. Проверьте, что в левой панели видна папка `work`, содержащая файлы из вашей текущей директории.
- Создайте новый Python-ноутбук (`File -> New -> Notebook`), выбрав ядро Python 3.
- В ячейке ноутбука выполните код для проверки окружения:

```
import numpy as np
import pandas as pd
print("NumPy version:", np.__version__)
print("Pandas version:", pd.__version__)
```

- Остановите контейнер, нажав **Ctrl+C** в терминале, где он запущен, или найдя его ID (`docker ps`) и выполнив `docker stop <container_id>`.

Критерии оценки для Части 1:

- **Удовлетворительно:** Успешно выполнены Этап 1 (установка Conda, создание окружения) и первые два пункта Этапа 2 (установка Docker, команда `hello-world`). Предоставлены соответствующие скриншоты.
- **Хорошо:** Дополнительно успешно выполнен пункт по запуску JupyterLab в контейнере и проверке работы ноутбука. Предоставлены скриншоты и даны ответы на контрольные вопросы.
- **Отлично:** Все задания выполнены в полном объеме. В ответах на вопросы продемонстрировано понимание принципов работы инструментов.

Рекомендуемая литература (источники)

1. **Официальная документация Conda:** <https://docs.conda.io/>
2. **Официальная документация Docker:** <https://docs.docker.com/>
3. **Jupyter Docker Stacks:** <https://jupyter-docker-stacks.readthedocs.io/>
4. **Ubuntu 24.04 Documentation:** <https://help.ubuntu.com/>
5. **Python Packaging User Guide:** <https://packaging.python.org/>
6. **Виртуальные окружения в Python:** <https://docs.python.org/3/tutorial/venv.html>
7. **Docker Get Started Tutorial:** <https://docs.docker.com/get-started/>
8. **Conda User Guide:** <https://docs.conda.io/projects/conda/en/latest/user-guide/>
9. **Статья "Зачем нужен Docker":** <https://habr.com/ru/company/ruvds/blog/438796/>
10. **Книга "Python for Data Analysis" by Wes McKinney:** (Главы о настройке окружения)